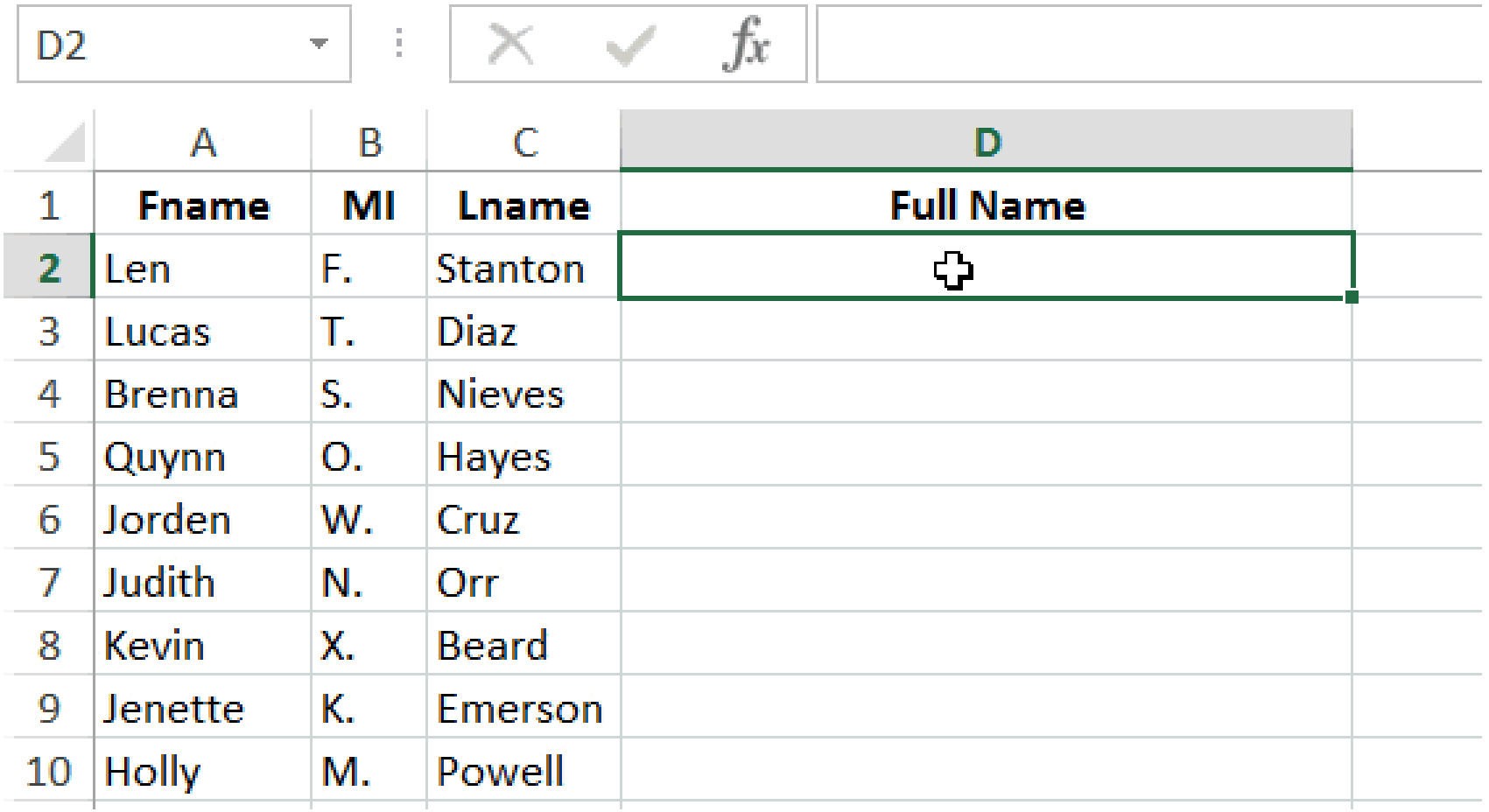
TaCLe

Learning constraints in spreadsheets and tabular data

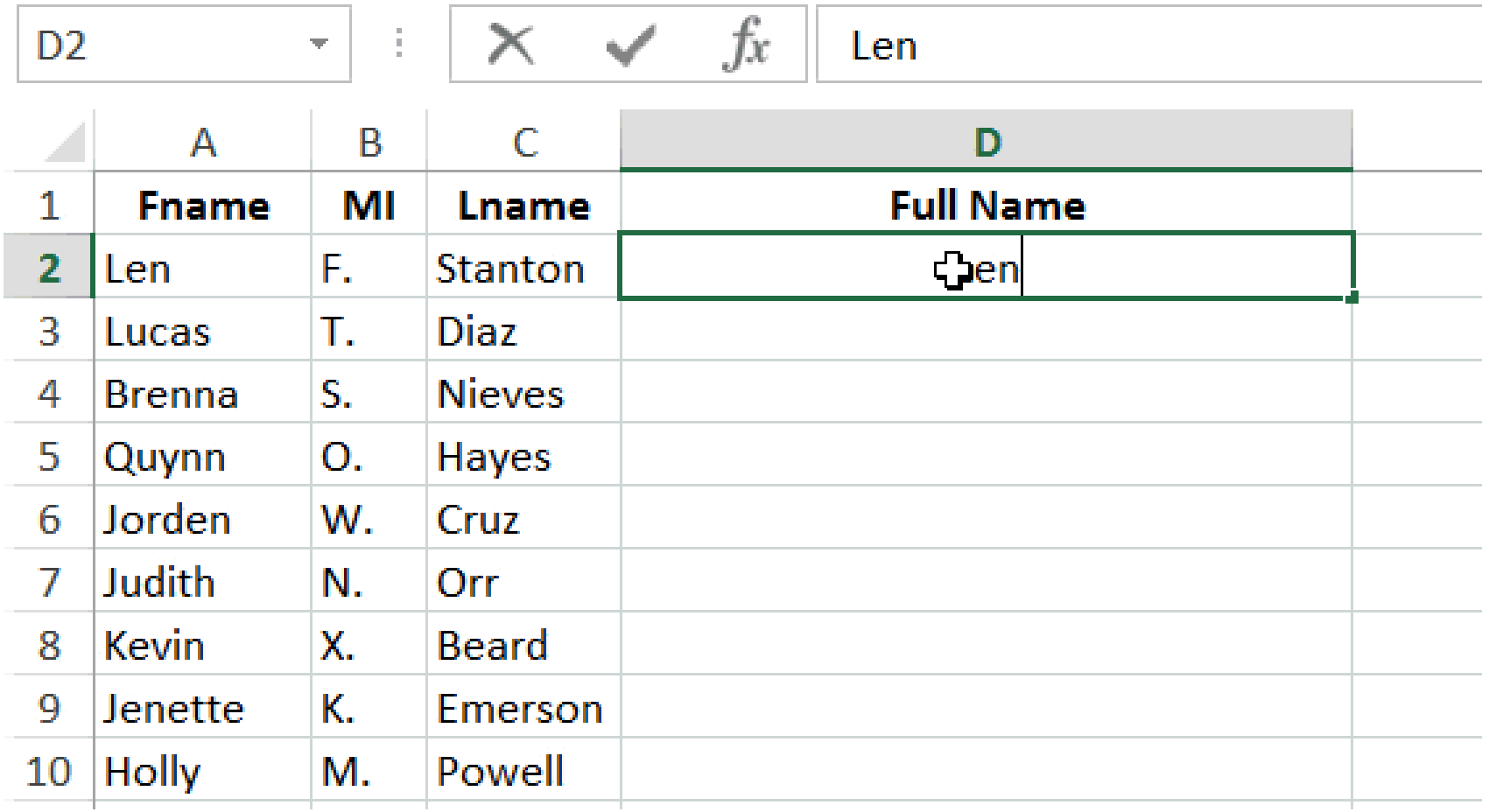
Samuel Kolb, Sergey Paramonov, Tias Guns, Luc De Raedt

September 6, 2017

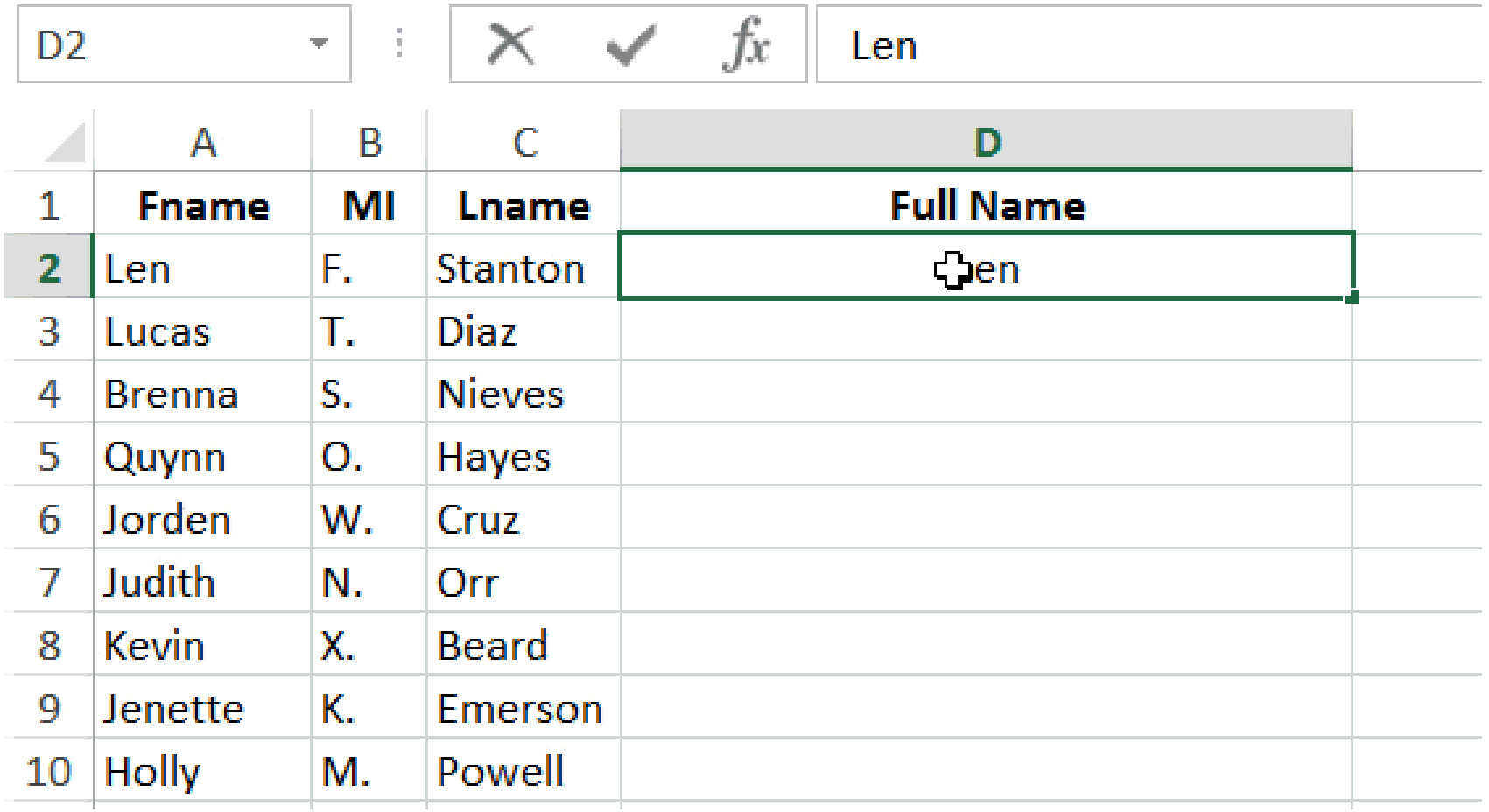
Inspiration: Flash-fill



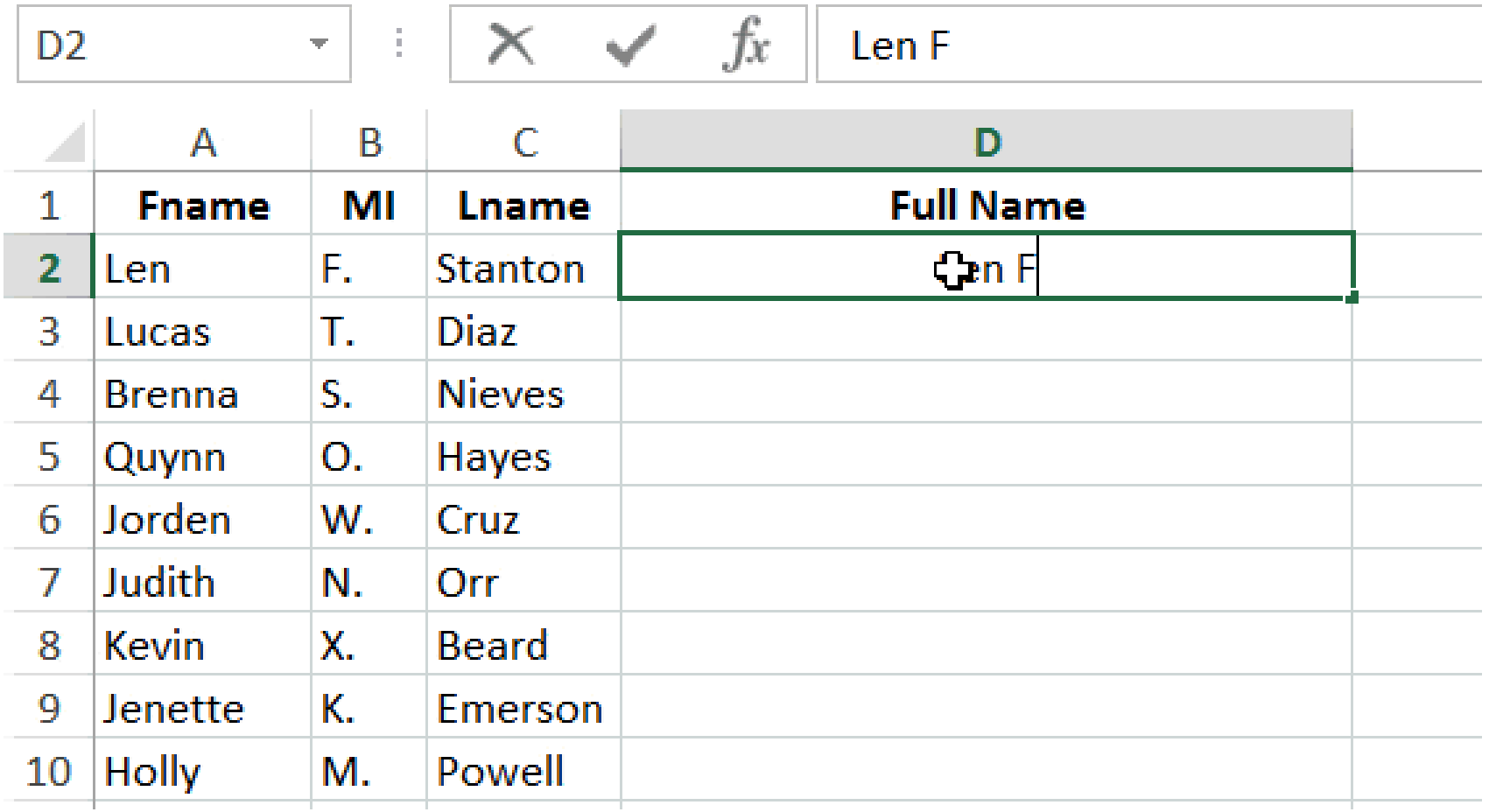
Inspiration: Flash-fill



Inspiration: Flash-fill

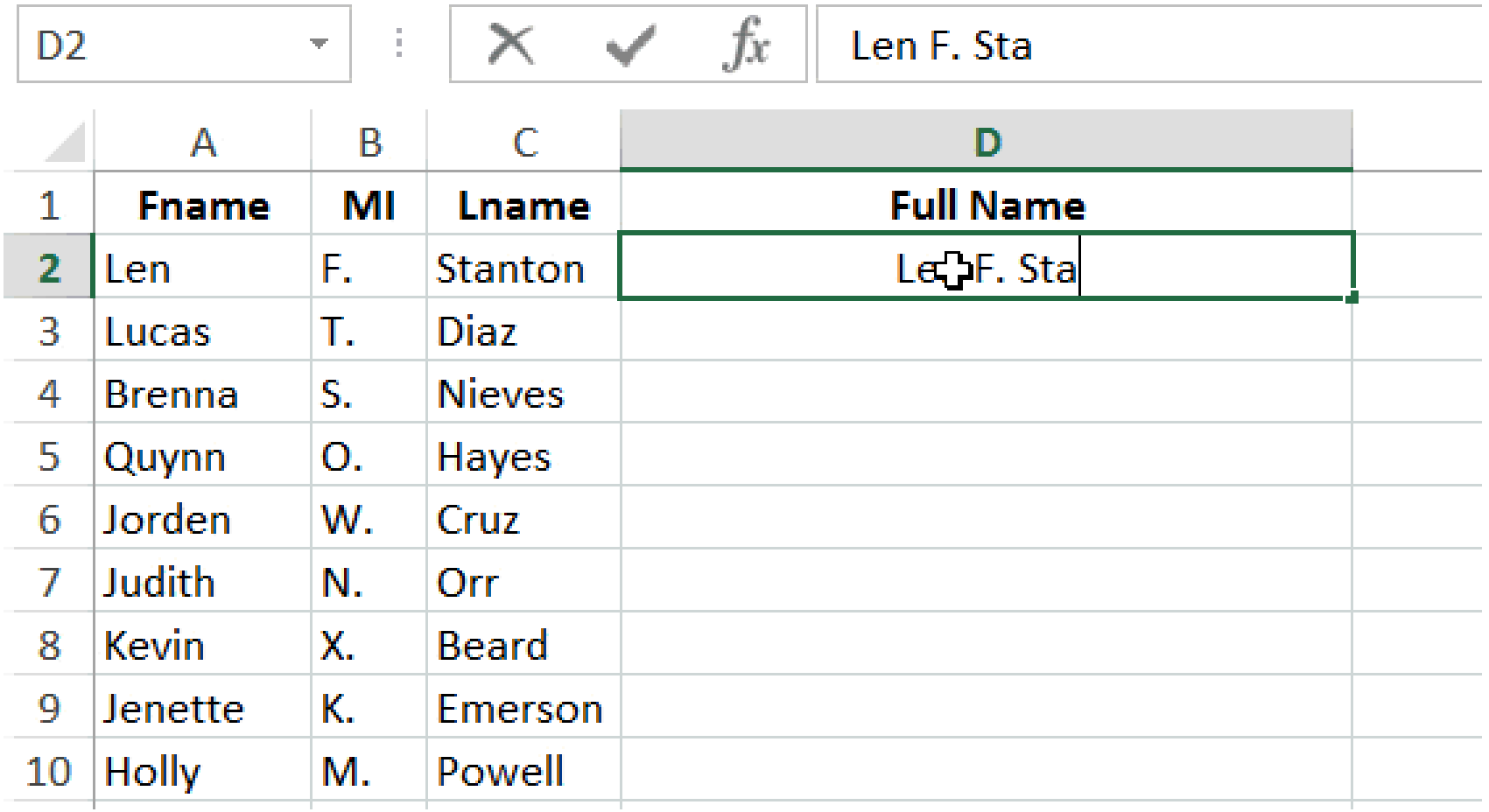


Inspiration: Flash-fill



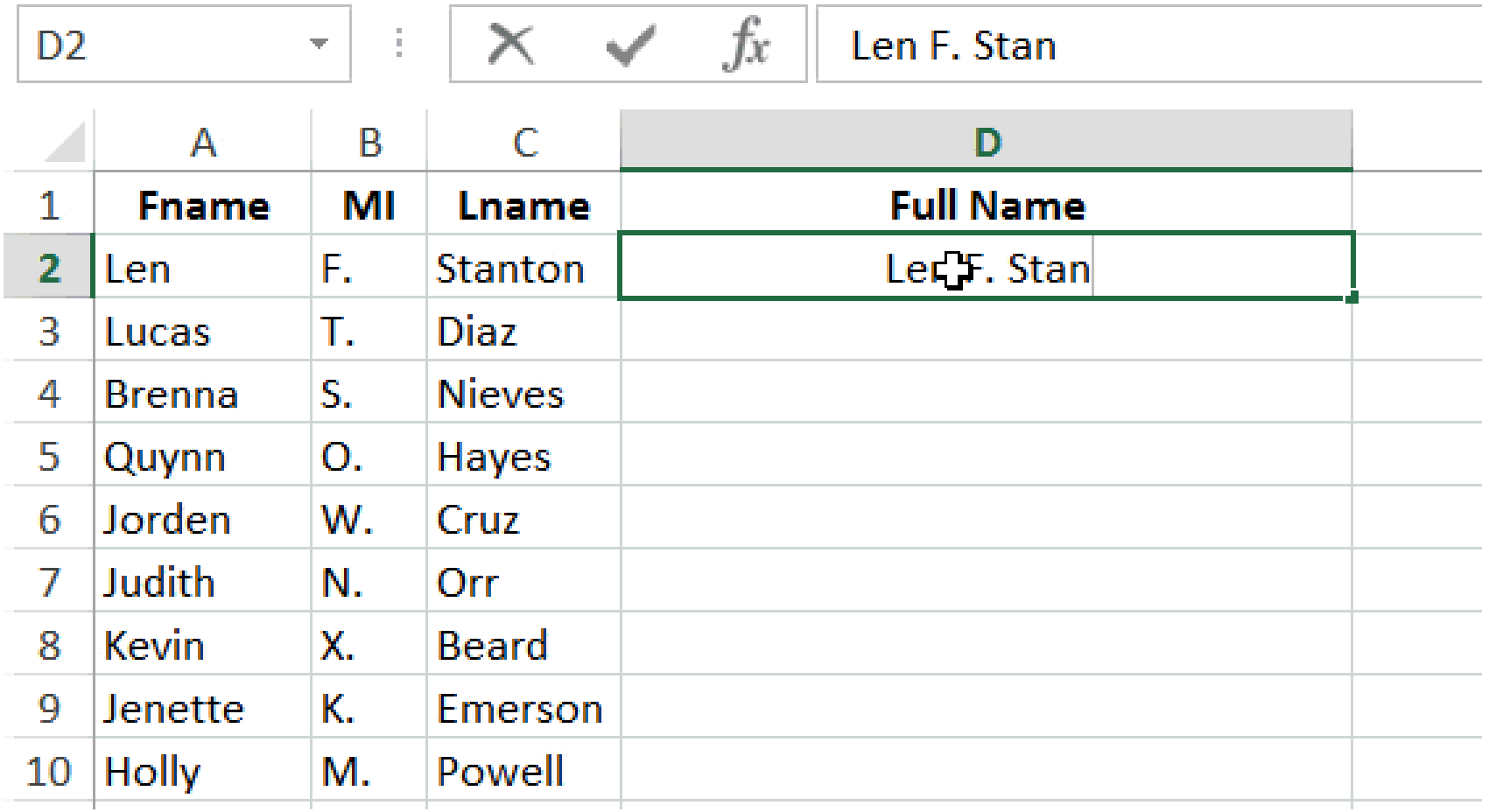
	A	B	C	D
1	**Fname**	**MI**	**Lname**	**Full Name**
2	Len	F.	Stanton	n F
3	Lucas	T.	Diaz	
4	Brenna	S.	Nieves	
5	Quynn	O.	Hayes	
6	Jorden	W.	Cruz	
7	Judith	N.	Orr	
8	Kevin	X.	Beard	
9	Jenette	K.	Emerson	
10	Holly	M.	Powell	

Inspiration: Flash-fill

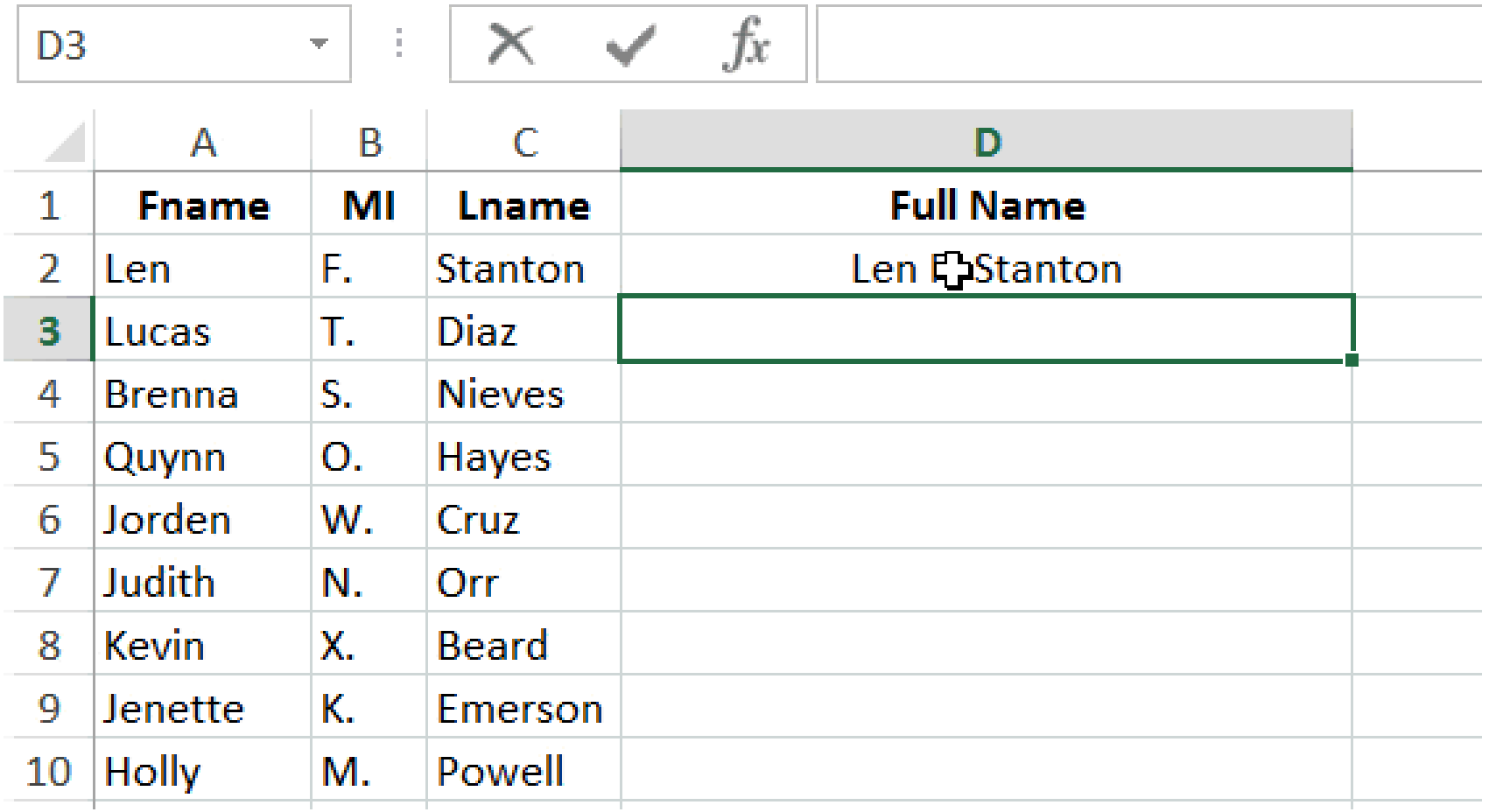


	A	B	C	D
1	**Fname**	**MI**	**Lname**	**Full Name**
2	Len	F.	Stanton	Len F. Sta
3	Lucas	T.	Diaz	
4	Brenna	S.	Nieves	
5	Quynn	O.	Hayes	
6	Jorden	W.	Cruz	
7	Judith	N.	Orr	
8	Kevin	X.	Beard	
9	Jenette	K.	Emerson	
10	Holly	M.	Powell	

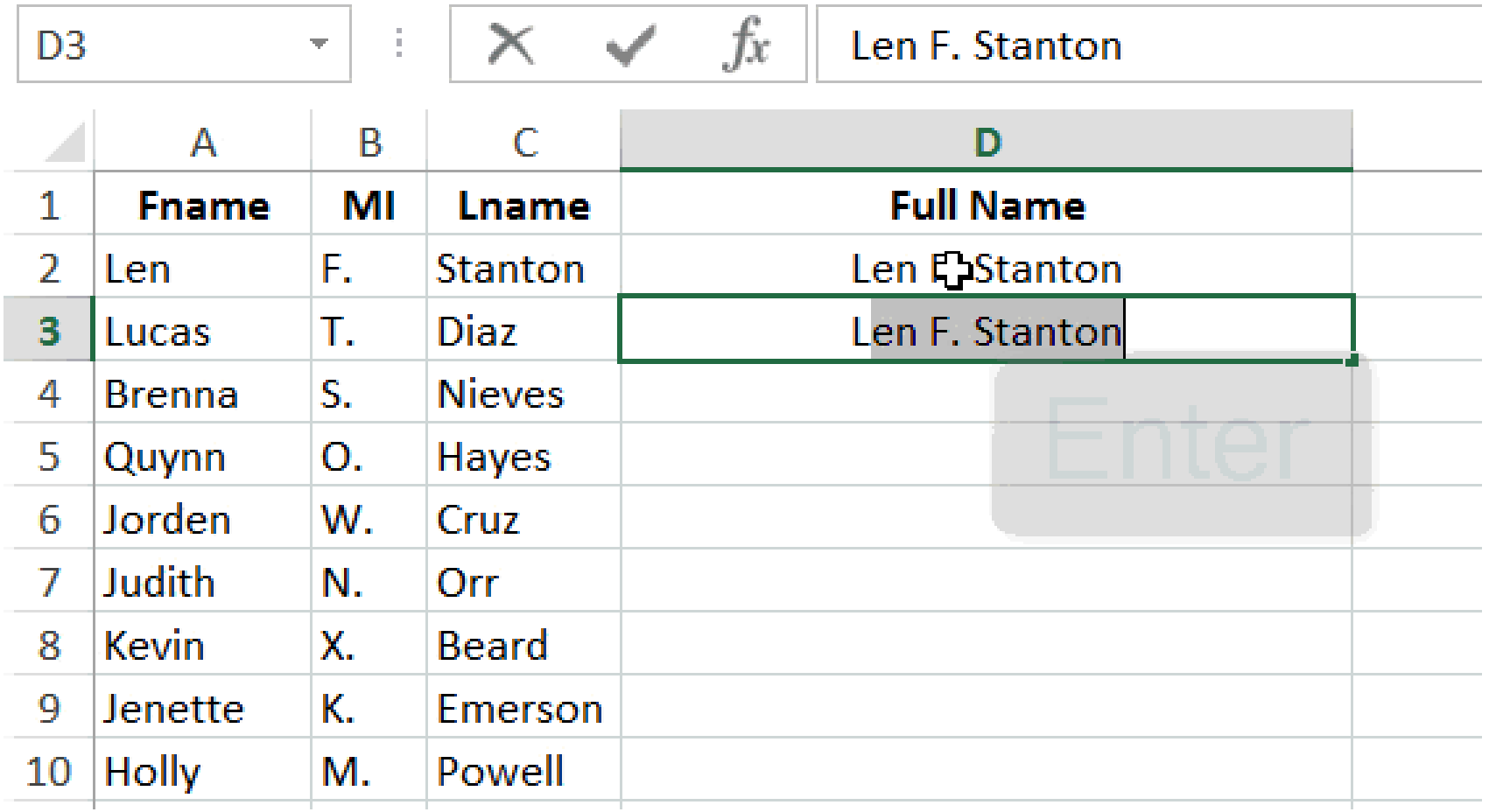
Inspiration: Flash-fill



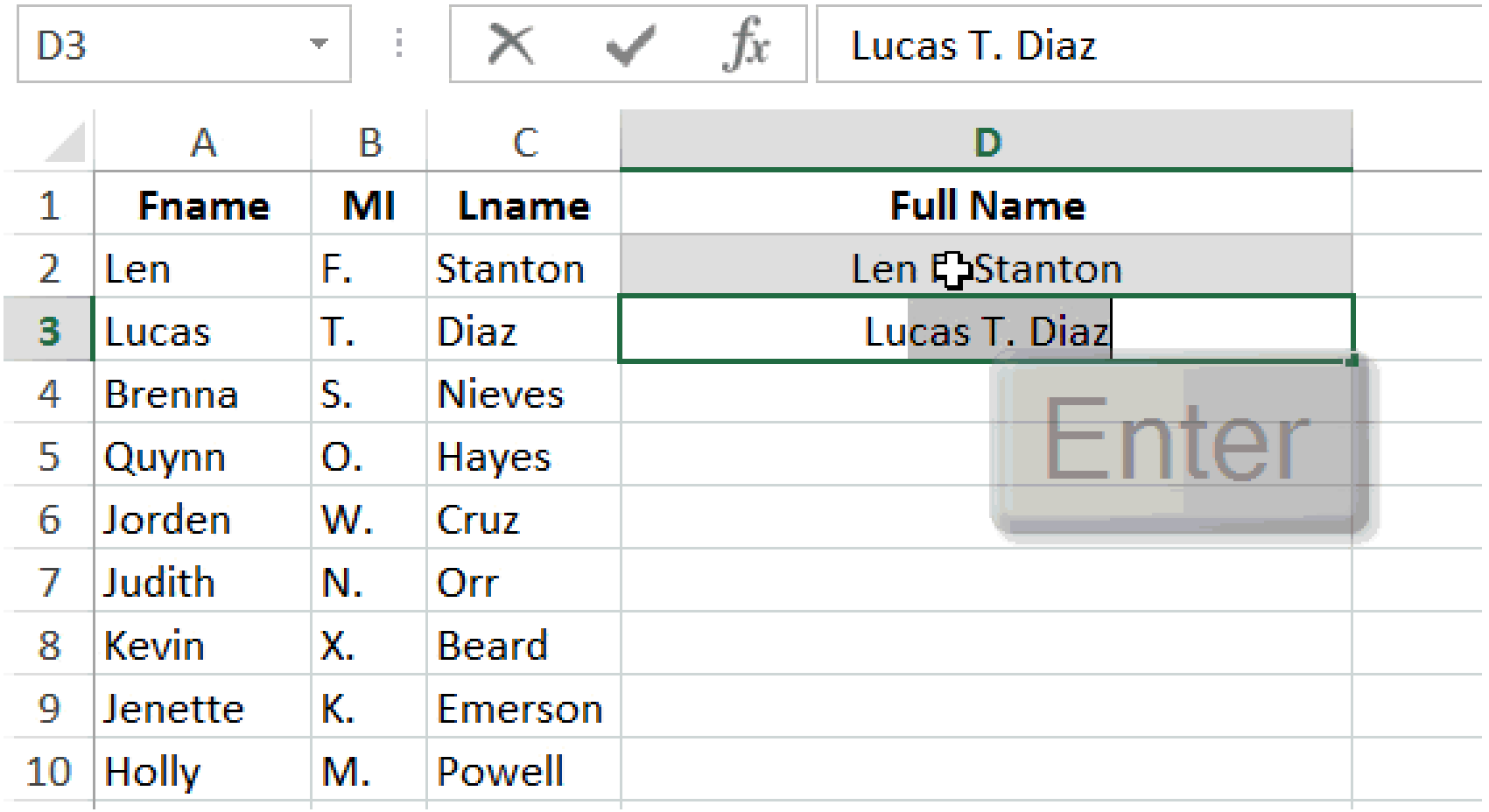
Inspiration: Flash-fill



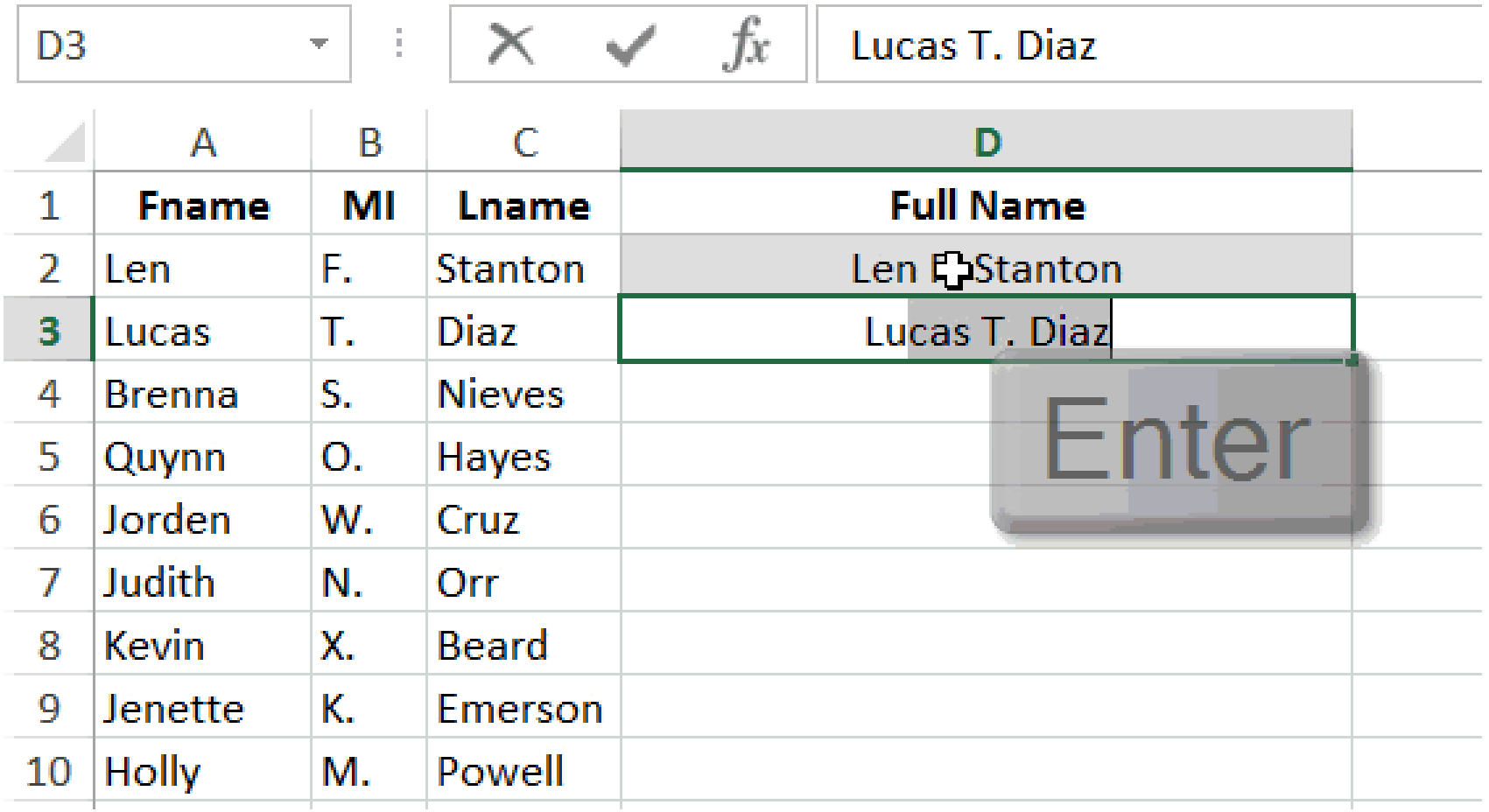
Inspiration: Flash-fill



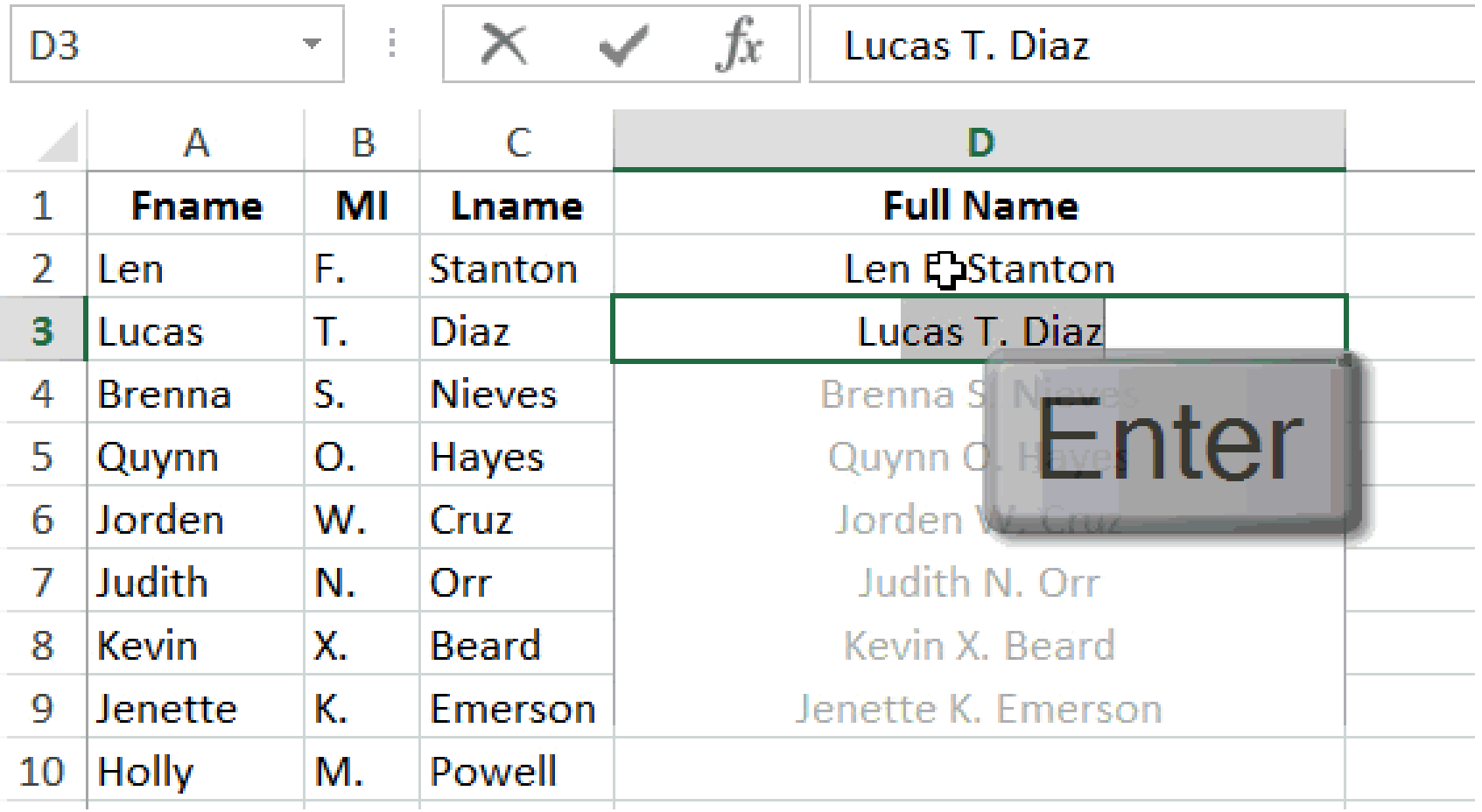
Inspiration: Flash-fill



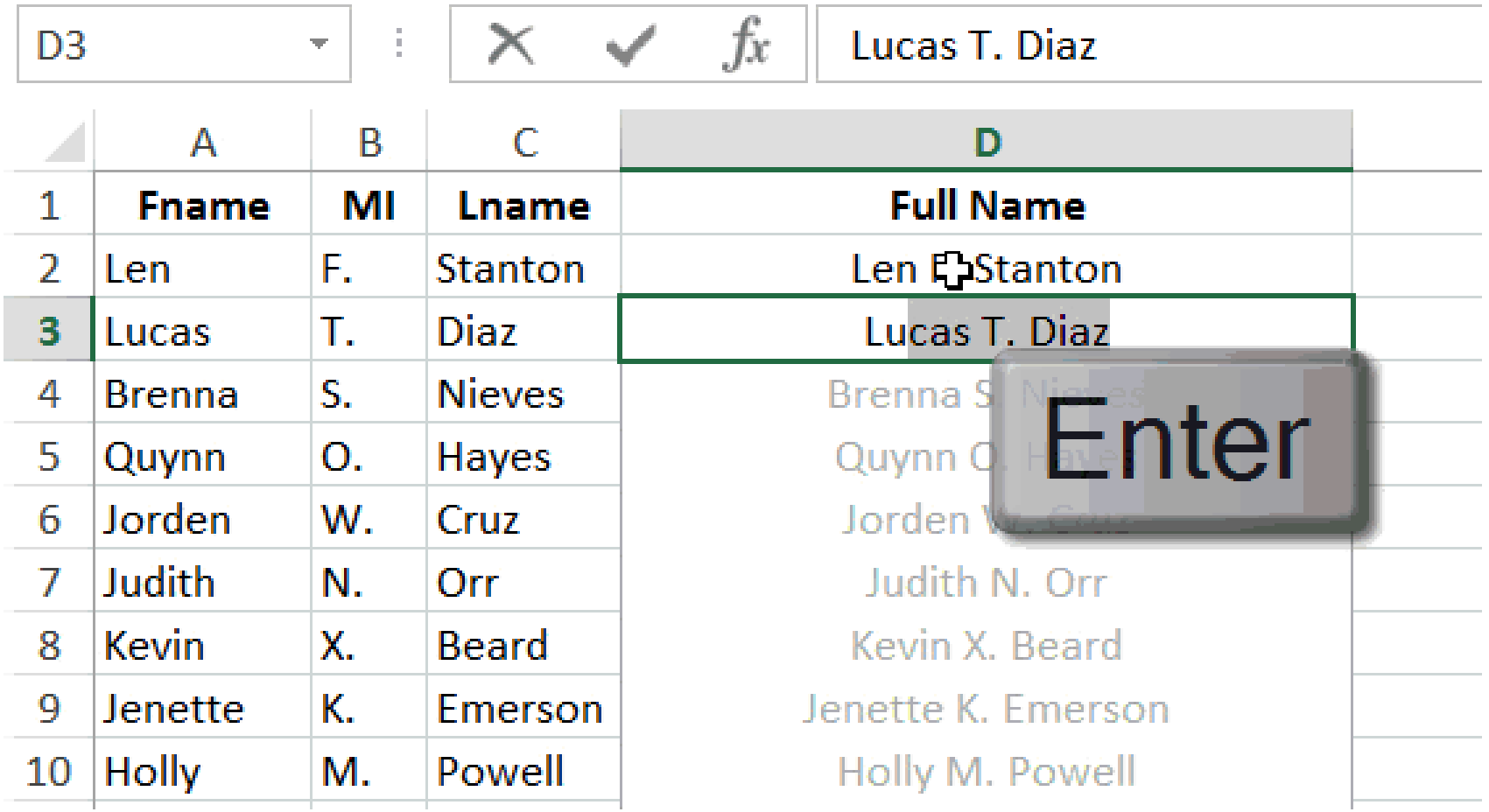
Inspiration: Flash-fill



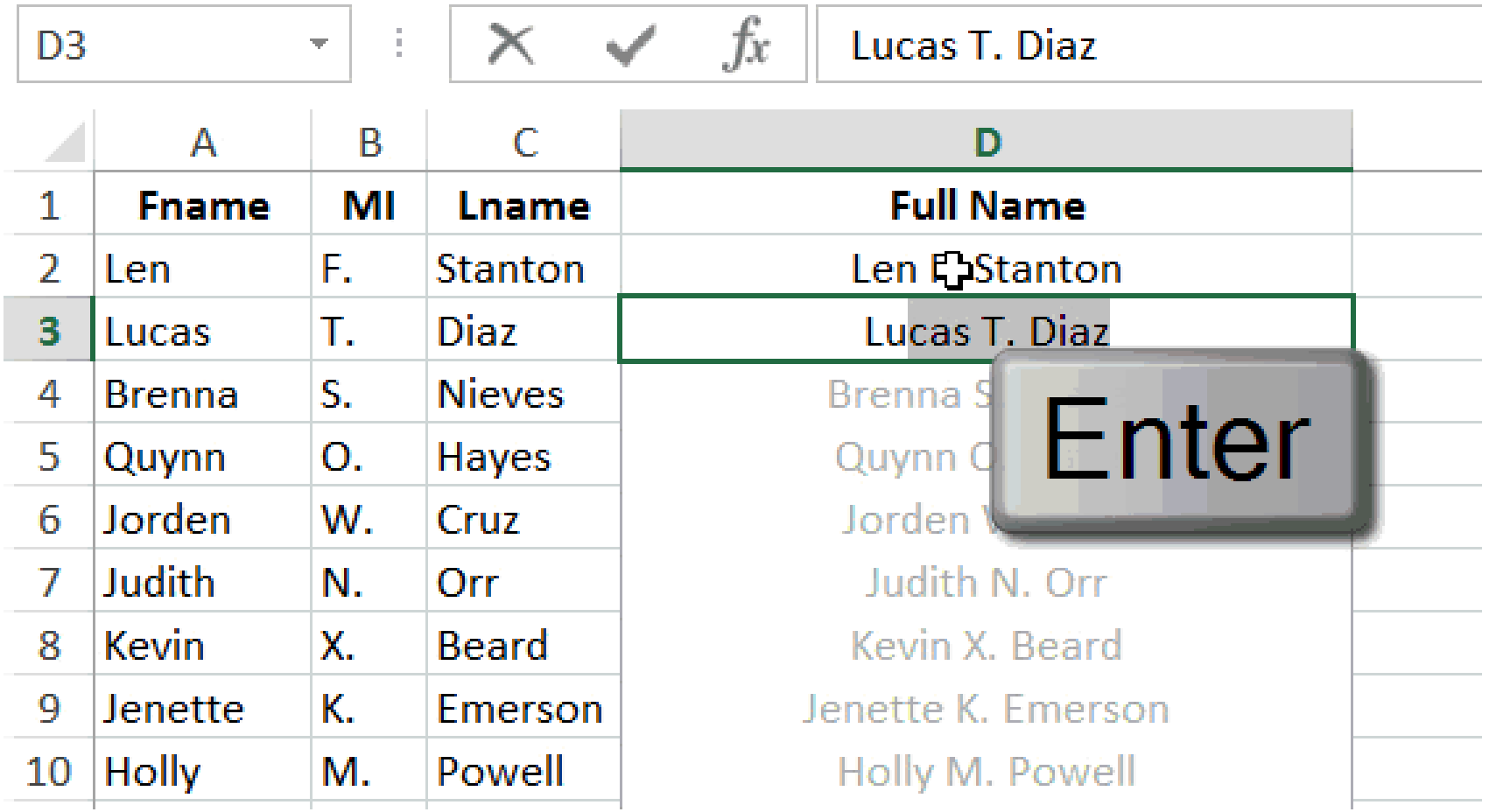
Inspiration: Flash-fill



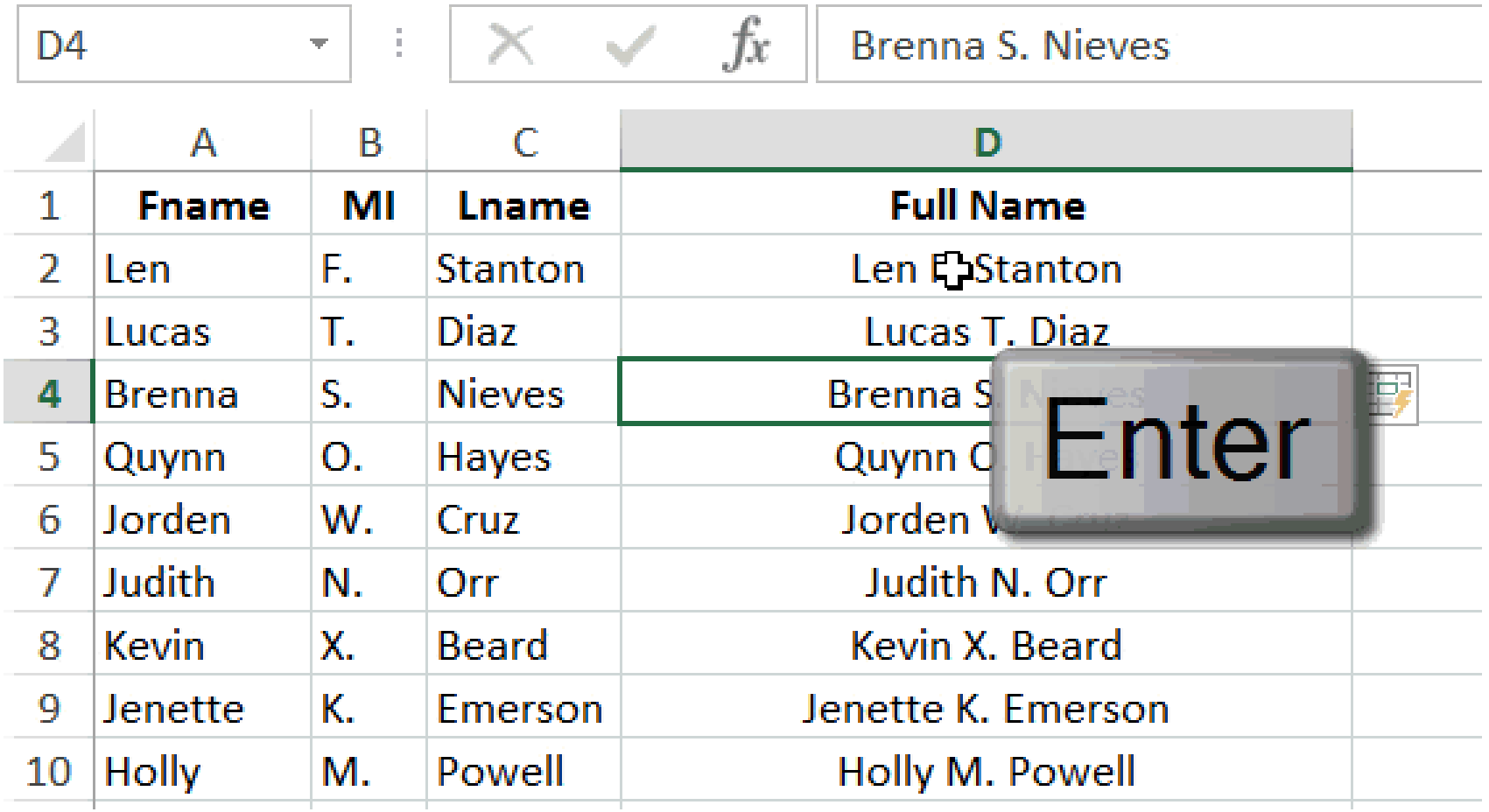
Inspiration: Flash-fill



Inspiration: Flash-fill

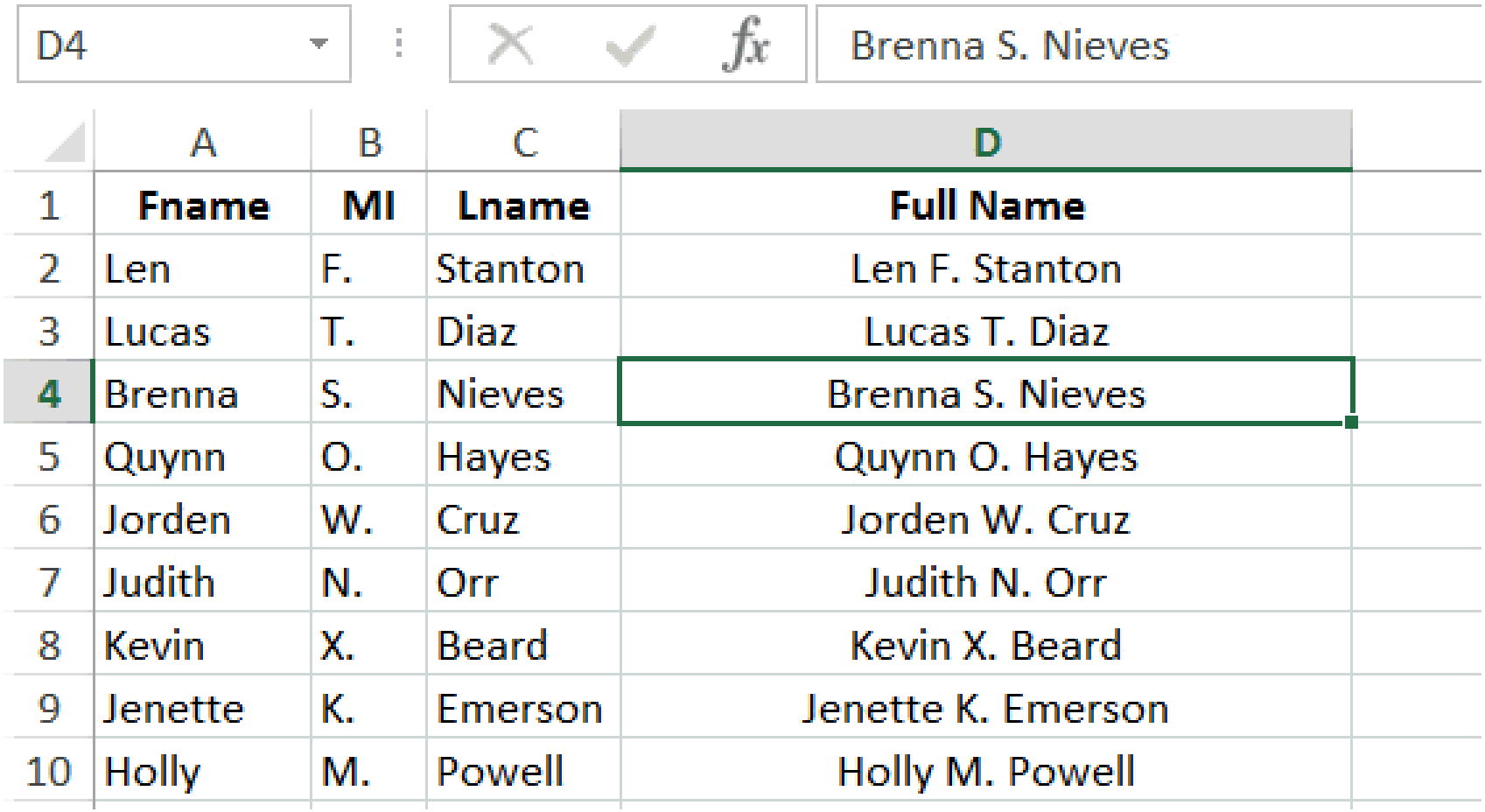


Inspiration: Flash-fill



	A	B	C	D
1	**Fname**	**MI**	**Lname**	**Full Name**
2	Len	F.	Stanton	Len F. Stanton
3	Lucas	T.	Diaz	Lucas T. Diaz
4	Brenna	S.	Nieves	Brenna S. Nieves
5	Quynn	O.	Hayes	Quynn O. Hayes
6	Jorden	W.	Cruz	Jorden W. Cruz
7	Judith	N.	Orr	Judith N. Orr
8	Kevin	X.	Beard	Kevin X. Beard
9	Jenette	K.	Emerson	Jenette K. Emerson
10	Holly	M.	Powell	Holly M. Powell

Inspiration: Flash-fill



	A	B	C	D
1	**Fname**	**MI**	**Lname**	**Full Name**
2	Len	F.	Stanton	Len F. Stanton
3	Lucas	T.	Diaz	Lucas T. Diaz
4	Brenna	S.	Nieves	Brenna S. Nieves
5	Quynn	O.	Hayes	Quynn O. Hayes
6	Jorden	W.	Cruz	Jorden W. Cruz
7	Judith	N.	Orr	Judith N. Orr
8	Kevin	X.	Beard	Kevin X. Beard
9	Jenette	K.	Emerson	Jenette K. Emerson
10	Holly	M.	Powell	Holly M. Powell

What is Flash-fill [Gulwani et al.]?

- learns string transformation
- classic ML supervised setting
- one positive (or very few) example
- integrated into Excel

Key Questions:

- what if we make it unsupervised?
- what if we learn constraints rather than functions?

Can we recover formulas from a CSV file?

Product	State	Sep	Oct	Nov	Sub-total		Product:	Sales:
Cherries	CA	200	360	230	790		Apples	2070
Bananas	AZ	350	230	150	730		Bananas	2420
Apples	TX	180	270	200	650		Lemons	740
Bananas	KS	400	240	310	950			
Lemons	AL	250	360	130	740			
Apples	FL	120	120	380	620			
Bananas	LA	330	270	140	740			
Apples	KY	110	320	370	800			

- What are the formulas here?

Can we recover formulas from a CSV file?

Product	State	Sep	Oct	Nov	Sub-total		Product:	Sales:
Cherries	CA	200	360	230	790		Apples	2070
Bananas	AZ	350	230	150	730		Bananas	2420
Apples	TX	180	270	200	650		Lemons	740
Bananas	KS	400	240	310	950			
Lemons	AL	250	360	130	740			
Apples	FL	120	120	380	620			
Bananas	LA	330	270	140	740			
Apples	KY	110	320	370	800			

- What are the formulas here?
- `T1[:, 6] = SUM(T1[:, 3:5], row)`

Can we recover formulas from a CSV file?

Product	State	Sep	Oct	Nov	Sub-total		Product:	Sales:
Cherries	CA	200	360	230	790		Apples	2070
Bananas	AZ	350	230	150	730		Bananas	2420
Apples	TX	180	270	200	650		Lemons	740
Bananas	KS	400	240	310	950			
Lemons	AL	250	360	130	740			
Apples	FL	120	120	380	620			
Bananas	LA	330	270	140	740			
Apples	KY	110	320	370	800			

- What are the formulas here?
- `T1[:, 6] = SUM(T1[:, 3:5], row)`
- `T2[:, 2] = SUMIF(T1[:, 1]=T2[:, 1], T1[:, 6])`

What is new here?

- Atypical data settings: no variables in columns and transactions in rows – everything mixed and position matters
- Multi-relational learning (across multiple tables, e.g., lookup)
- Constraints – specific for spreadsheets (e.g., fuzzy lookup, sumproduct)
- Important details:
  - *None* values
  - Semi-structured data
  - Numeric and textual data
  - Numerical precision

Formalization

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

T1

B1 = T1[:, 1] B2 = T1[:, 2] B3 = T1[:, 3:8] B4 = T1[:, 9] B5 = T1[:, 10:11]

T2

Total	991	1030	1046	1081	4148
Average	247.75	257.5	261.5	270.25	1037
Max	370	408	396	387	1551
Min	93	98	96	105	392

B6 = T2[1:4, :]

Salesperson	Items sold
Diana Coolen	5
Marc Desmet	10
Marc Desmet	8
Diana Coolen	9
Birgit Kenis	15
Marc Desmet	8
Birgit Kenis	2
Diana Coolen	20
Marc Desmet	3
Kris Goossens	19

T3

B8 = T3[:, 1] B9 = T3[:, 2]

Quarter	Income	Expenses	Total
Q1	991	212	779
Q2	1030	710	1099
Q3	1046	137	2008
Q4	1081	240	2849

T4

B10 = T4[:, 1] B11 = T4[:, 2:4]

Customer	Contact	Contact Name
Frank	1	Diana Coolen
Sarah	3	Kris Goossens
George	3	Kris Goossens
Mary	2	Diana Coolen
Tim	4	Birgit Kenis

T5

B12 = T5[:, 1] B13 = T5[:, 2] B14 = T5[:, 3]

Tables (T)

- $n \times m$ matrix
- Headerless
- (Optional: orientation)

Formalization

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

T1

B1 = T1[:, 1] B2 = T1[:, 2] B3 = T1[:, 3:8] B4 = T1[:, 9] B5 = T1[:, 10:11]

T2

Total	991	1030	1046	1081	4148
Average	247.75	257.5	261.5	270.25	1037
Max	370	408	396	387	1551
Min	93	98	96	105	392

B6 = T2[1:4, :]

Salesperson	Items sold
Diana Coolen	5
Marc Desmet	10
Marc Desmet	8
Diana Coolen	9
Birgit Kenis	15
Marc Desmet	8
Birgit Kenis	2
Diana Coolen	20
Marc Desmet	3
Kris Goossens	19

T3

B8 = T3[:, 1] B9 = T3[:, 2]

Quarter	Income	Expenses	Total
Q1	991	212	779
Q2	1030	710	1099
Q3	1046	137	2008
Q4	1081	240	2849

T4

B10 = T4[:, 1] B11 = T4[:, 2:4]

Customer	Contact	Contact Name
Frank	1	Diana Coolen
Sarah	3	Kris Goossens
George	3	Kris Goossens
Mary	2	Diana Coolen
Tim	4	Birgit Kenis

T5

B12 = T5[:, 1] B13 = T5[:, 2] B14 = T5[:, 3]

Blocks (B)

- Contiguous group of entire rows or entire columns (vectors)
- Type-consistent
- Fixed orientation

Formalization

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

T1

B1 = T1[:, 1] B2 = T1[:, 2] B3 = T1[:, 3:8] B4 = T1[:, 9] B5 = T1[:, 10:11]

T2

Total	991	1030	1046	1081	4148
Average	247.75	257.5	261.5	270.25	1037
Max	370	408	396	387	1551
Min	93	98	96	105	392

B6 = T2[1:4, :]

Salesperson	Items sold
Diana Coolen	5
Marc Desmet	10
Marc Desmet	8
Diana Coolen	9
Birgit Kenis	15
Marc Desmet	8
Birgit Kenis	2
Diana Coolen	20
Marc Desmet	3
Kris Goossens	19

T3

B8 = T3[:, 1] B9 = T3[:, 2]

Quarter	Income	Expenses	Total
Q1	991	212	779
Q2	1030	710	1099
Q3	1046	137	2008
Q4	1081	240	2849

T4

B10 = T4[:, 1] B11 = T4[:, 2:4]

Customer	Contact	Contact Name
Frank	1	Diana Coolen
Sarah	3	Kris Goossens
George	3	Kris Goossens
Mary	2	Diana Coolen
Tim	4	Birgit Kenis

T5

B12 = T5[:, 1] B13 = T5[:, 2] B14 = T5[:, 3]

Block containment ($B' \sqsubseteq B$)

- B' subblock of B
- B superblock of B'
- Supports different granularities for reasoning

Formalization

Block properties

type	a block is type-consistent, so it has one type
table	the table that the block belongs to
orientation	either row-oriented or column-oriented
size	the number of vectors a block contains
length	the length of its vectors; as all vectors are from the same table, they always have the same length;
rows	the number of rows in the block; in row-oriented blocks this is equivalent to the size;
columns	the number of columns in the block; in row-oriented blocks this is equivalent to the length.

Formalization

Constraint templates

- Syntax
  - Syntactic form, e.g. $ALLDIFFERENT(B_x)$
  - In logic: relation / predicate
- Signature
  - Requirements on arguments, e.g. $discrete(B_x)$
  - In logic: bias (Sig_s)
- Definition
  - Actual definition, e.g. $i \neq j$: $B_x[i] \neq B_x[j]$
  - In logic: ackground knowledge (Def_s)

Formalization

Row-wise sum

- Syntax: $B_r = SUM_{row}(\mathbf{B_x})$
- Signature: B_r and $\mathbf{B_x}$ are *numeric*; $columns(\mathbf{B_x}) \geq 2$; and $rows(\mathbf{B_x}) = length(B_r)$
- Definition: $B_r[i] = \sum_{j=1}^{columns(\mathbf{B}_x)} row(i, \mathbf{B}_x)[j]$

Formalization

Row-wise sum

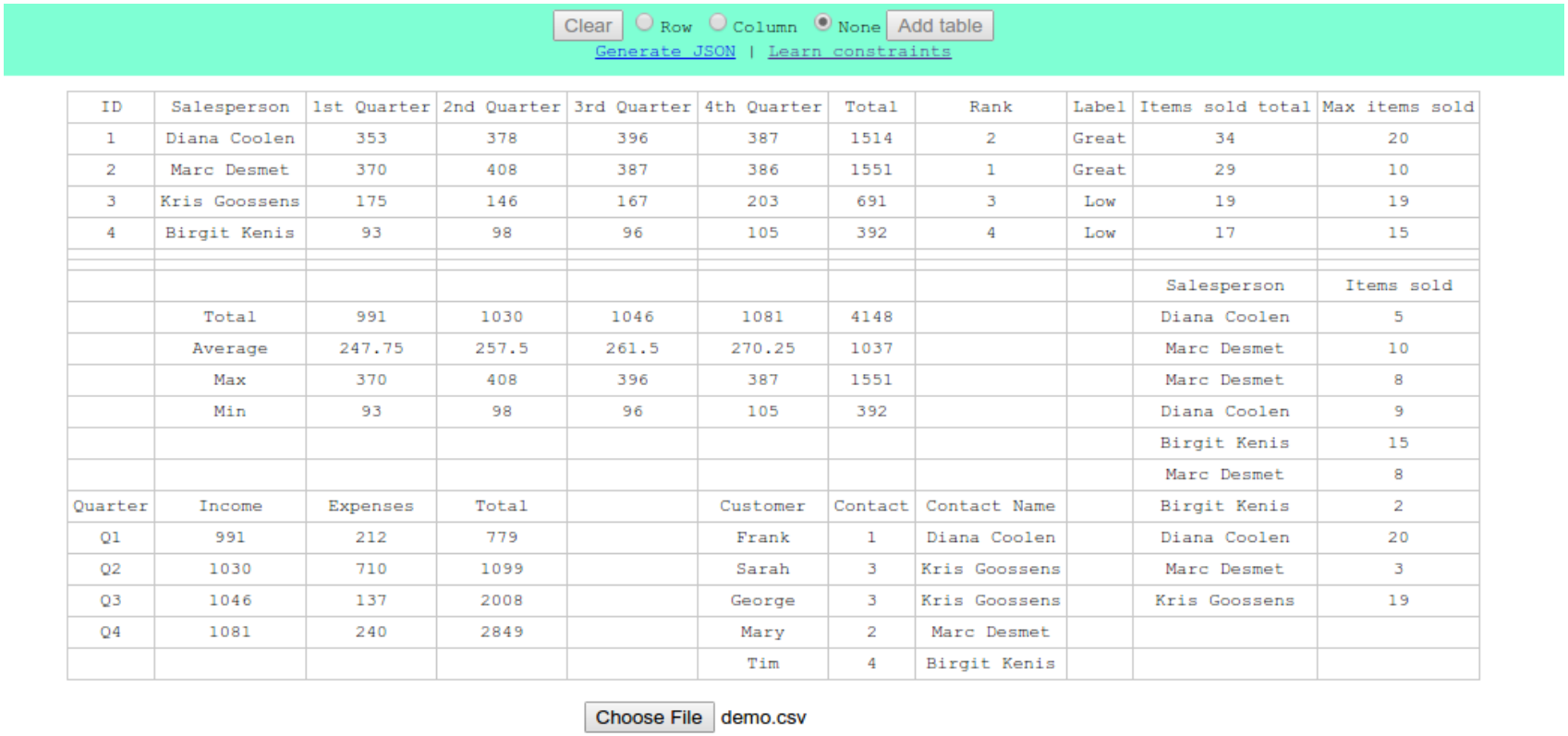
- Syntax: $B_r = SUM_{row}(\mathbf{B_x})$
- Signature: B_r and $\mathbf{B_x}$ are *numeric*; $columns(\mathbf{B_x}) \geq 2$; and $rows(\mathbf{B_x}) = length(B_r)$
- Definition: $B_r[i] = \sum_{j=1}^{columns(\mathbf{B}_x)} row(i, \mathbf{B}_x)[j]$

Lookup

- Syntax: $B_r = LOOKUP(B_{fk}, B_{pk}, B_{val})$
- Signature: B_{fk} and B_{pk} are *discrete*; arguments $\{B_{fk}, B_r\}$ and $\{B_{pk}, B_{val}\}$ within the same set have the same *length*, *table* and *orientation*; B_r and B_{val} have the same type; and $FOREIGNKEY(B_{fk}, B_{pk})$.
- Definition: $B_r[i] = B_{val}[j]$ where $B_{pk}[j] = B_{fk}[i]$

High level approach

1. Detect tables (Visual selection)



ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

									Salesperson	Items sold
	Total	991	1030	1046	1081	4148			Diana Coolen	5
	Average	247.75	257.5	261.5	270.25	1037			Marc Desmet	10
	Max	370	408	396	387	1551			Marc Desmet	8
	Min	93	98	96	105	392			Diana Coolen	9
									Birgit Kenis	15
									Marc Desmet	8

Quarter	Income	Expenses	Total		Customer	Contact	Contact Name		Birgit Kenis	2
Q1	991	212	779		Frank	1	Diana Coolen		Diana Coolen	20
Q2	1030	710	1099		Sarah	3	Kris Goossens		Marc Desmet	3
Q3	1046	137	2008		George	3	Kris Goossens		Kris Goossens	19
Q4	1081	240	2849		Mary	2	Marc Desmet			
					Tim	4	Birgit Kenis			

High level approach

1. Detect tables (Visual selection)

Clear Row Column None Add table

Generate JSON | Learn constraints

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

									Salesperson	Items sold
	Total	991	1030	1046	1081	4148			Diana Coolen	5
	Average	247.75	257.5	261.5	270.25	1037			Marc Desmet	10
	Max	370	408	396	387	1551			Marc Desmet	8
	Min	93	98	96	105	392			Diana Coolen	9
									Birgit Kenis	15
									Marc Desmet	8

Quarter	Income	Expenses	Total		Customer	Contact	Contact Name		Birgit Kenis	2
Q1	991	212	779		Frank	1	Diana Coolen		Diana Coolen	20
Q2	1030	710	1099		Sarah	3	Kris Goossens		Marc Desmet	3
Q3	1046	137	2008		George	3	Kris Goossens		Kris Goossens	19
Q4	1081	240	2849		Mary	2	Marc Desmet			
					Tim	4	Birgit Kenis			

Choose File demo.csv

High level approach

1. Detect tables (Visual selection)

Clear Row Column None Add table

Generate JSON | Learn constraints

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

									Salesperson	Items sold
	Total	991	1030	1046	1081	4148			Diana Coolen	5
	Average	247.75	257.5	261.5	270.25	1037			Marc Desmet	10
	Max	370	408	396	387	1551			Marc Desmet	8
	Min	93	98	96	105	392			Diana Coolen	9
									Birgit Kenis	15
									Marc Desmet	8

Quarter	Income	Expenses	Total		Customer	Contact	Contact Name		Birgit Kenis	2
Q1	991	212	779		Frank	1	Diana Coolen		Diana Coolen	20
Q2	1030	710	1099		Sarah	3	Kris Goossens		Marc Desmet	3
Q3	1046	137	2008		George	3	Kris Goossens		Kris Goossens	19
Q4	1081	240	2849		Mary	2	Marc Desmet			
					Tim	4	Birgit Kenis			

Choose File demo.csv

High level approach

2. Detect blocks (Automatically, transparant to the user)

ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

T1

B1 = T1[:, 1] B2 = T1[:, 2] B3 = T1[:, 3:8] B4 = T1[:, 9] B5 = T1[:, 10:11]

T2

Total	991	1030	1046	1081	4148
Average	247.75	257.5	261.5	270.25	1037
Max	370	408	396	387	1551
Min	93	98	96	105	392

B6 = T2[1:4, :]

Salesperson	Items sold
Diana Coolen	5
Marc Desmet	10
Marc Desmet	8
Diana Coolen	9
Birgit Kenis	15
Marc Desmet	8
Birgit Kenis	2
Diana Coolen	20
Marc Desmet	3
Kris Goossens	19

T3

B8 = T3[:, 1] B9 = T3[:, 2]

Quarter	Income	Expenses	Total
Q1	991	212	779
Q2	1030	710	1099
Q3	1046	137	2008
Q4	1081	240	2849

T4

B10 = T4[:, 1] B11 = T4[:, 2:4]

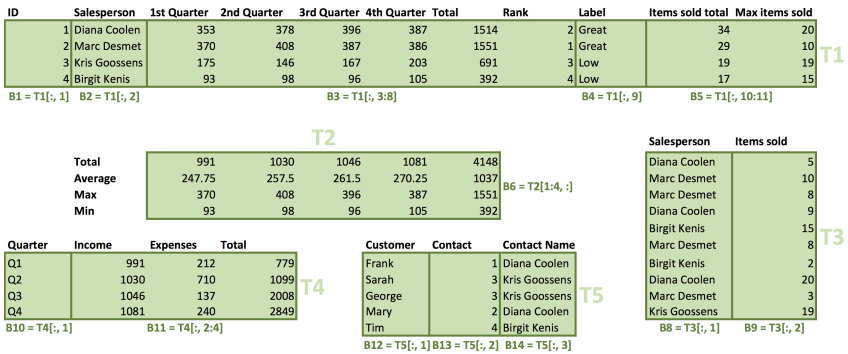
Customer	Contact	Contact Name
Frank	1	Diana Coolen
Sarah	3	Kris Goossens
George	3	Kris Goossens
Mary	2	Diana Coolen
Tim	4	Birgit Kenis

T5

B12 = T5[:, 1] B13 = T5[:, 2] B14 = T5[:, 3]

High level approach

3. Learn constraints



ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank	Label	Items sold total	Max items sold
1	Diana Coolen	353	378	396	387	1514	2	Great	34	20
2	Marc Desmet	370	408	387	386	1551	1	Great	29	10
3	Kris Goossens	175	146	167	203	691	3	Low	19	19
4	Birgit Kenis	93	98	96	105	392	4	Low	17	15

T1

B1 = T1[:, 1] B2 = T1[:, 2] B3 = T1[:, 3:8] B4 = T1[:, 9] B5 = T1[:, 10:11]

T2

Total	991	1030	1046	1081	4148
Average	247.75	257.5	261.5	270.25	1037
Max	370	408	396	387	1551
Min	93	98	96	105	392

B6 = T2[1:4, :]

Salesperson	Items sold
Diana Coolen	5
Marc Desmet	10
Marc Desmet	8
Diana Coolen	9
Birgit Kenis	15
Marc Desmet	8
Birgit Kenis	2
Diana Coolen	20
Marc Desmet	3
Kris Goossens	19

T3

B8 = T3[:, 1] B9 = T3[:, 2]

Quarter	Income	Expenses	Total
Q1	991	212	779
Q2	1030	710	1099
Q3	1046	137	2008
Q4	1081	240	2849

T4

B10 = T4[:, 1] B11 = T4[:, 2:4]

Customer	Contact	Contact Name
Frank	1	Diana Coolen
Sarah	3	Kris Goossens
George	3	Kris Goossens
Mary	2	Diana Coolen
Tim	4	Birgit Kenis

T5

B12 = T5[:, 1] B13 = T5[:, 2] B14 = T5[:, 3]

$$SERIES(T_1[:,1])$$

$$T_1[:,1] = RANK(T_1[:,5])^*$$

$$T_1[:,1] = RANK(T_1[:,6])^*$$

$$T_1[:,1] = RANK(T_1[:,10])^*$$

$$T_1[:,8] = RANK(T_1[:,7])$$

$$T_1[:,8] = RANK(T_1[:,3])^*$$

$$T_1[:,8] = RANK(T_1[:,4])^*$$

$$T_1[:,7] = SUM_{row}(T_1[:,3{:}6])$$

$$T_1[:,10] = SUMIF(T_3[:,1], T_1[:,2], T_3[:,2])$$

$$T_1[:,11] = MAXIF(T_3[:,1], T_1[:,2], T_3[:,2])$$

$$T_2[1,:] = SUM_{col}(T_1[:,3{:}7])$$

$$T_2[2,:] = AVERAGE_{col}(T_1[:,3{:}7])$$

$$T_2[3,:] = MAX_{col}(T_1[:,3{:}7]),$$

$$T_2[4,:] = MIN_{col}(T_1[:,3{:}7])$$

$$T_4[:,2] = SUM_{col}(T_1[:,3{:}6])$$

$$T_4[:,4] = PREV(T_4[:,4]) + T_4[:,2] - T_4[:,3]$$

$$T_5[:,2] = LOOKUP(T_5[:,3], T_1[:,2], T_1[:,1])^*$$

$$T_5[:,3] = LOOKUP(T_5[:,2], T_1[:,1], T_1[:,2])$$

Approach

Intuition

- Example: $Y = SUM_{col}(X)$
- Step 1: Find superblock assignments (i.e. suitable blocks)
  - Assignments *compatible* signature, relax where necessary
  - e.g. $numeric(X)$, $numeric(Y)$, $columns(X) \geq 2$
  - e.g. $rows(X) = length(Y)$
- Step 2: Find all constraints over subblocks of the superblock assignments
  - Find *subassignments* that satisfy the signature and the definition
  - e.g. find subblocks for X and Y such that $Y[i] = \sum column\ i$

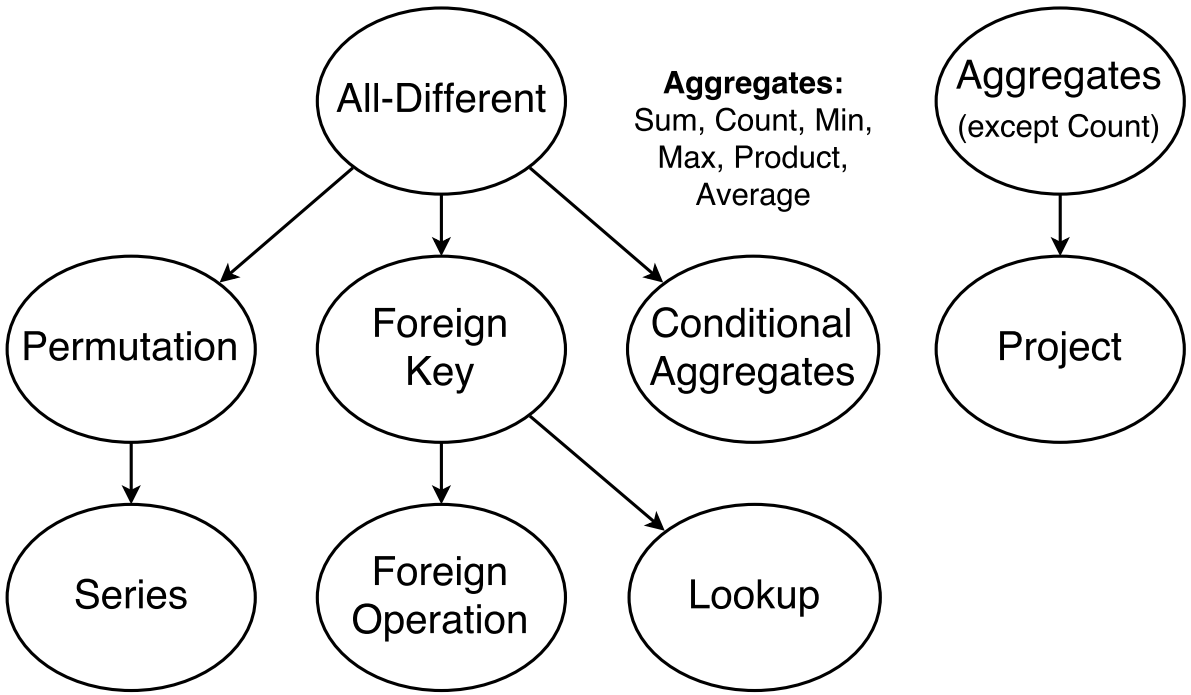
Approach

Refinements (similar as in constraint learning - clausal discovery)

- Dependencies between constraints
- Redundancies in the output constraints
- Limited precision

Refinements

Dependencies



Reuse learned constraints to learn dependent constraints

Refinements

Redundancies

- Hide constraints that are *equivalent*
- Equivalent constraints can be computed from one another
- Example: $B_1 = B_2 \times B_3$ and $B_1 = B_3 \times B_2$
- $\rightarrow$ Use canonical form

Limited precision

- Use the *result* value to deduce required precision
- Compute formula on input values and round to precision

Evaluation

Evaluation questions

- Recall (*Q1*)
- Precision (*Q2*)
- Speed (*Q3*)

Evaluation

Method

- Benchmark spreadsheets collected
  - Exercise session
  - Online tutorials
  - *Data* spreadsheets (e.g. crime statistics, fincancial data)

Evaluation

Method

- Benchmark spreadsheets collected
  - Exercise session
  - Online tutorials
  - *Data* spreadsheets (e.g. crime statistics, fincancial data)
- Convert all spreadsheets to CSV files

Evaluation

Method

- Benchmark spreadsheets collected
  - Exercise session
  - Online tutorials
  - *Data* spreadsheets (e.g. crime statistics, fincancial data)
- Convert all spreadsheets to CSV files
- Manually specify ground-truth: *intended* constraints
  - Based on original formulas, context, headers (intuition)
  - Structural constraints are ignored

Evaluation

Benchmark

	Exercises (9)		**Tutorials** (21)		**Data** (4)	
	Overall	Sheet avg	Overall	Sheet avg	Overall	Sheet avg
Tables	19	2.11	48	2.29	4	1
Cells	1231	137	1889	90	2320	580
Intended Constraints	34	3.78	52	2.48	6	1.50

Evaluation

Benchmark

	Exercises (9)		**Tutorials** (21)		**Data** (4)	
	Overall	Sheet avg	Overall	Sheet avg	Overall	Sheet avg
Recall	0.85	0.83	0.88	0.87	1.00	1.00
Recall Supported	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.97	0.98	0.70	0.91	1.00	1.00
Speed (s)	1.62	0.18	1.76	0.08	0.81	0.20

Evaluation

Benchmark

	Exercises (9)		**Tutorials** (21)		**Data** (4)	
	Overall	Sheet avg	Overall	Sheet avg	Overall	Sheet avg
Recall	0.85	0.83	0.88	0.87	1.00	1.00
Recall Supported	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.97	0.98	0.70	0.91	1.00	1.00
Speed (s)	1.62	0.18	1.76	0.08	0.81	0.20

- Q1. How many intended constraints are found by *TaCLe*?
  - High recall
  - All supported constraints always found

Evaluation

Benchmark

	Exercises (9)		**Tutorials** (21)		**Data** (4)	
	Overall	Sheet avg	Overall	Sheet avg	Overall	Sheet avg
Recall	0.85	0.83	0.88	0.87	1.00	1.00
Recall Supported	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.97	0.98	0.70	0.91	1.00	1.00
Speed (s)	1.62	0.18	1.76	0.08	0.81	0.20

- Q1. How many intended constraints are found by *TaCLe*?
  - High recall
  - All supported constraints always found
- Q2. How precise is *TaCLe*?
  - Precise on most spreadsheets
  - Duplicates / multiple ways to calculate thwart precision

Evaluation

Benchmark

	Exercises (9)		**Tutorials** (21)		**Data** (4)	
	Overall	Sheet avg	Overall	Sheet avg	Overall	Sheet avg
Recall	0.85	0.83	0.88	0.87	1.00	1.00
Recall Supported	1.00	1.00	1.00	1.00	1.00	1.00
Precision	0.97	0.98	0.70	0.91	1.00	1.00
Speed (s)	1.62	0.18	1.76	0.08	0.81	0.20

- Q1. How many intended constraints are found by *TaCLe*?
  - High recall
  - All supported constraints always found
- Q2. How precise is *TaCLe*?
  - Precise on most spreadsheets
  - Duplicates / multiple ways to calculate thwart precision
- Q3. How fast is *TaCLe*?
  - Dependencies crucial

Smart import

TaCLe + constraint translation + cycle breaking

Clear ○ Row ○ Column ◉ None Add table

Generate JSON | Learn constraints

☐ A2:A5 = RANK(E2:E5) | ☐ A2:A5 = RANK(F2:F5) | ☑ SERIES(A2:A5)

☑ G2:G5 = SUM(C2:F5, row)

☐ H2:H5 = RANK(C2:C5) | ☐ H2:H5 = RANK(D2:D5) | ☑ H2:H5 = RANK(G2:G5)

☑ C8:G8 = SUM(C2:G5, col)

☑ C9:G9 = AVERAGE(C2:G5, col)

☑ C10:G10 = MAX(C2:G5, col)

☑ C11:G11 = MIN(C2:G5, col)

☐ C14:C18 = LOOKUP(D14:D18, B2:B5, A2:A5) | ☑ D14:D18 = LOOKUP(C14:C18, A2:A5, B2:B5)

	A	B	C	D	E	F	G	H
1	ID	Salesperson	1st Quarter	2nd Quarter	3rd Quarter	4th Quarter	Total	Rank
2	1	Diana Coolen	365	378	396	387	1526	2
3	2	Marc Desmet	370	408	387	386	1551	1
4	3	Kris Goossens	175	146	167	203	691	3
5	4	Birgit Kenis	93	98	96	105	392	4
6								
7								
8		Total	1003	1030	1046	1081	4160	
9		Average	250.75	257.5	261.5	270.25	1040	
10		Max	370	408	396	387	1551	
11		Min	93	98	96	105	392	
12								
13		Customer	Contact	Contact Name				
14		Frank	1	Diana Coolen				
15		Sarah	3	Kris Goossens				
16		George	3	Kris Goossens				
17		Mary	2	Marc Desmet				
18		Tim	4	Birgit Kenis				

Auto-completion / dynamic error checking

incremental TaCLe + vector tracking

Date	Region	Units	Total Amt.
24/06/12	North	186	$50,592.00
01/06/12	East	356	$96,832.00
09/09/12	West	907	$246,704.00
26/06/12	South	190	$51,680.00
22/04/12	North	717	$195,024.00
22/03/12	West	550	$149,600.00
19/12/11	East	942	$256,224.00
31/10/11	North	901	$245,072.00
02/10/11	West	117	$31,824.00

T1

B1 = T1[:, 1:2] B2 = T1[:, 3:4]

Region	Units per region	Amt. per region
B3 = T2[:, 1]	B4 = T2[:, 2:3]	
North	1804	490688
East	1298	353056
South	190	51680
West	*1574*	*428128*
ΔD	***Suggested***	***Suggested***

T2

Conclusion

- Approach that learns constraints in spreadsheet
  - Accurate
  - Rather precise
  - Efficient

Future work

- Applications (incremental learning, vector tracking, noise)
- Nested constraints
- Sub vector-size
- Post-processing (heuristic / entailment)

Questions?

https://unsplash.com/search/photos/cat?photo=Qmox1MkYDnY

Questions?